



DECUS

PROGRAM LIBRARY

DECUS NO.	8-491
TITLE	INDEXED FLOATING POINT MATH SUBROUTINES FOR PDP-8/E
AUTHOR	Jacques Ricard
COMPANY	National Film Board of Canada Montreal, Canada
DATE	October 28, 1971
SOURCE LANGUAGE	PAL III

Although this program has been tested by the contributor, no warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related program material, and no responsibility is assumed by these parties in connection therewith.

DECUS



1900

INDEXED FLOATING POINT MATH SUBROUTINES FOR PDP-8/E

DECUS Program Library Write-up

DECUS NO. 8-491

DESCRIPTION: This package of subroutines, which may be assembled in any order and anywhere in 4K, uses a PDP-8/e with EAE option in B mode. It contains various subroutines to perform floating point math, input/output, logical functions and other house keeping routines.

These subroutines fall in three basic categories:

- 1- indexed addressing
- 2- direct addressing
- 3- non addressing

Discussing them in order;

INDEXED ADDRESSING

General Calling Sequence:

```
CALL JMS SSS           /call of indexed subroutine
CALL +1 xxxx          /offset
```

Upon entry the contents of the AC is saved on PAGE ZERO in PSAC. The address of the operand is calculated by adding the offset to the contents for the register labelled INDEX. This address is recorded in OPAD. Then the subroutine function SSS is performed with the operand and contents of FAC then control is returned to CALL+2.

FAC means floating accumulator (HORD, LORD, EXP).

Standard floating point notation is defined as:

LOC contains low order mantissa (12BITS).

LOC +1 contains high order signed mantissa (12BITS).

LOC +2 contains signed exponent.

Floating point indexed subroutine

FPGET	Gets a floating point number from the operand and stores it in the FAC. The previous content of the FAC is lost. Return: AC=0 MQ=not changed L=1
FPPUT	Saves content of FAC in operand. FAC not changed. Previous operand lost. Return: AC=0 MQ=not changed L=1
FPZER	Clear a floating point operand. FAC not changed. Return: AC=0 MQ=0 L=undefined.
FPADD	Adds the content of the FAC to the content of the operand and places the normalized result in the FAC. Return: AC=0 MQ, L are undefined.
FPSUB	Subtracts in floating point the contents of the operand from the contents of the FAC and places the normalized result in the FAC. Return: AC=0 MQ, L are undefined.

.../

FPMUJ Multiplies the contents of the FAC by the contents of the operand in floating point. Places the normalized result in the FAC.

Return: AC=0 MQ, L are undefined.

FPDIV Divides in floating point the contents of the FAC by the contents of the operand and places the normalized result in the FAC.

Return: AC=0 MQ, L are undefined.

Logical indexed subroutines

XCLOR	Takes the exclusive or between the contents of the AC and the contents of the operand (single precision)
-------	--

Return: AC contains result, MQ=0, L is undefined.

INCOR	Takes the inclusive or between the content of the AC and the contents of the operand. (single precision)
-------	--

Return: AC contains result, MQ operand, L is undefined.

Misc. indexed subroutines

INJMS Executes a JMS indirect via the contents of the operand.
 (usually used as an indexed JMS indirect)

Enters subroutine AC, MQ are unchanged, L is undefined.

INDCA Deposits contents of AC in operand.

Return: AC=0, MQ is not changed, L is undefined.

DIRECT ADDRESSING SUBROUTINES

General calling sequence:

	/AC contains address of operand
JMS DDD	/call of subroutine

Performs function DDD with the operand whose address is in the AC.

FADA Adds in floating point the contents of the FAC to the contents of the operand. The normalized result of placed in the FAC.

The content of the operand is not changed.

Return: AC=0, MQ, L are undefined.

FSUBA Subtracts in floating point the contents of the operand from the contents of the FAC. The normalized result is placed in the FAC. The contents of the operand is not changed.

Return: AC=0, MQ, L are undefined.

FLMY Multiplies in floating point the contents of the FAC by the contents of the operand. The normalized result is placed in the FAC. The contents of the operand is not changed.

Return: AC=0, MQ, L are undefined.

...

FLDV Floating point divide of contents of FAC by contents of operand.
The result is left in the FAC. The contents of the operand is
not changed.
Return: AC=0, MQ, L are undefined.

The following direct addressing subroutines do not follow the general
calling sequence.

STORE Calling sequence:

 AC contains source address
CALL JMS STORE
CALL 1 destination address

Function: transfers three words from source to destination. The
contents of the source is not changed.

Return: AC=0, MQ is not changed, L=1

XORD Calling sequence:

 AC contains bits to be ored
 JMS XORD
 address of operand (single precision)

Function: exclusivly or's the content of the AC with the content
of the operand.

Return: AC contains result, MQ=0, L is undefined.

INCORD Calling sequence:

 AC contains bits to be ored
 JMS INCORD
 address of operand (single precision)

Function: inclusibly or's the content of the operand with the
content of the AC.

Return: AC contains result, MQ contains previous AC, L is not
changed.

NON ADDRESSING SUBROUTINES

Calling sequence 1 word
parameters may be in AC or FAC or somewhere else.

FPFLOA Floats converts to standard floating point notation a signed,
double precision number in AC, MQ into FAC.

FPFIX Round-off and fix integer FAC to a double precision, signed
number in AC, MQ.

.../

FPABS Gets the absolute value of the content of the FAC in the FAC.
Return: AC,MQ=0, L=1 if FAC was pos., otherwise L=0.

FPCOM Complements the content of the FAC.
Return: AC,MQ=0, L=1 if content of FAC is 0, otherwise L=1.

FPSQRT Calculates square root of the absolute content of the FAC.
Places the result in the FAC.
Return: AC=0, MQ, L are undefined

INCDX Adds 3 to the content of INDEX.
Return: AC,L=0, MQ is not changed.

INSEN Enter with the LIMIT in the AC. Then 3 is added to the INDEX
and INDEX is compared against LIMIT; if INDEX is exceeding
LIMIT then it returns call plus 1, otherwise return to call plus 2.
Return: AC=0, MQ=LIMIT, L is undefined.

FLINPT Inputs a signed decimal number and converts it to standard
floating point notation in the FAC. Uses CINPT for character
entry.
Return: AC=0, MQ, L are undefined. DIGC is 0 if there is no
numeric input otherwise, DIGC is non 0.

CINPT Inputs ASCII character to TERM from TTY.
Ignores code 212 and 252.
Accepts as legal codes 255 to 332, all others are classified
as terminators. If DIGC is 0 it will ignore terminators.
Returns to call plus 1 if a RUBOUT (377) is typed.
Returns to call plus 2 with DIGC incremented if a legal cha-
racter is typed in or a terminator is accepted.
Returns via BAKAR if back-arrow is typed.
Returns: AC=0, MQ is not changed, L is undefined, character
in term.

FLOUTP Outputs floating point contents of FAC as a signed, decimal
number according to a format specification held in FORM. Most
significant six bits specify the total number of digits to be
outputted. Least significant six bits specify the number of
digits right of the decimal point. If this number is 0 then
the decimal point is inhibited. Also an adequate round off
is made. If the format is too small then X's are typed.
Return: AC=0, MQ, L, FAC are undefined.

TYPE Outputs an ASCII character on TTY and echos a line feed after
each return.
Return: AC=0, MQ is not changed and L is undefined.

TENPOW Calculates in floating point a power of 10 as specified in
signed AC. Leaves normalized power in FAC.
Return: AC=0, MQ, L are undefined.

MISC. SUBROUTINES (see listings for more details)

GETOPA	Gets indexed operand address in OPAD and AC.
GETOP	Gets operand in operand storage on page 0.
ALIGN	Aligns operand and FAC exponents for add.
SWAP	Interchanges FAC and operand.
DIV1	Shift FAC mantissa right.
DIV2	Shift operand mantissa right.
DUNORM	Shift FAC and operand mantissas right.
ACMINS	Triple precision two's complement of FAC mantissa.
MINUS2	Triple precision two's complement of operand mantissa.
DNORM	Triple precision normalize FAC.
SIGN	Calculate the product of signs.
DMULT	Double precision multiply.
DPDIV	Double precision divide.
MULT10	Multiply FAC mantissa by 10 (decimal).

RESTRICTIONS:

When executing any of the subroutines the EAE option must have been initialized to the B mode instruction set.

